European Parliament Data Information Retrieval System

Catarina Figueiredo
University of Porto
up201606334@fe.up.pt

João Bernardo Sousa
University of Porto
up201606649@fe.up.pt

Mariana Dias
University of Porto
up201606486@fe.up.pt

Tiago Ribeiro
University of Porto
up201605619@fe.up.pt

ABSTRACT

In this article we ~~will~~ explain how the data preparation process of our information retrieval system on European Parliament data works. We will address every step in the process from dataset gathering, data preparation, natural-language processing and data storage in a relational database. Our goal is to analyze a variety of data related to the European Parliament in order for it to be more accessible to the general public.

KEYWORDS

Information Retrieval, European Parliament, Data Preparation

ACM Reference Format:
Catarina Figueiredo, João Bernardo Sousa, Mariana Dias, and Tiago Ribeiro. 2020. European Parliament Data Information Retrieval System. In *DAPI*. FEUP, Porto, Portugal, 4 pages.

1 INTRODUCTION

Over the years the volume of information has been growing and become accessible to all via the World Wide Web. The organization and structuring of information can bring many benefits to users, such as the dissemination of knowledge and relevant data. However, the emergent information has steadily become more unorganized and sparse making its research an arduous task.

Likewise the information provided by the European Parliament is very sparse, making it hard to analyse and search relevant information. For example, it is impossible to understand the trajectory of each European Parliament member. How did their work, voting behaviour and topics of interest change over time? Can we understand how political groups and individual politicians vote depending on the topic being voted, who submitted the proposal and other relevant factors?

For this reason, we decided to organize this data and make it available in a new interface with more advanced search parameters and new ways of visualizing possible patterns and other useful information.

2 DATASET PREPARATION

In this section we will go through the entire process related to data extraction and preparation for further usage, including the tools used, as can be seen in Figure 1.

2.1 Data Collection

For this project we used two sources: **Parltrack** and the **European Parliament website**.

Regarding the European Parliament website, we employed scraping techniques, such as HTML parsing, to extract relevant text data from **reports**, e.g. its text, rapporteur(s) and committee, if any.

As for Parltrack, it is an European initiative that aggregates information from various official EU sources and releases it in JSON format. It provides a huge amount of data so we decided to focus our efforts in a subset of the dumps provided, namely **Members of the European Parliament (MEPs)** and **MEP Plenary Votes**.

The MEPs dump contains information on all the current and previous members of the European Parliament since 2004, including their names, age, country, political groups affiliation, national party affiliation, committees they were or are part of and their social media info, while the MEP Plenary Votes dump contains information on the votes cast by MEPs in the plenary (in favor, against or abstention) and information on what is being voted.

2.2 Data preparation

In this stage we used **OpenRefine**, a specialized tool for data cleaning. We started by normalized the data provided by the *Parltrack* dumps. Political group names' were normalized, since some of them were abreviated and others not, and the names refering to the same group were often in different languages. Insertion errors in the committees names were also fixed. These errors were mostly related with the arbitrary usage of single and double quotes interchangeably. Finally, we removed data that was not relevant for our project. The dumps including much more information that are not relevant for the goals of this project, for example the *Curriculum Vitae* of the MEPs. We also greatly reduced the number of plenary votes because we decided to include only final votes on final proposals, removing all the votes on amendments and paragraph changes.

The **Parltrack** JSON dumps contain for each plenary vote only some basic information regarding the corresponding resolution. On the other hand, the **EP** website contains the full text of each document, complete with the author and respective committee. In order to cross these two data sources, some **Python** libraries where used: For each vote in the **Parltrack** dump, the respective document code was used to access the EP website URL and download the page with the content. Then, by using the **PyQuery** library, the HTML code was parsed. Because the documents' pages don't follow a common structure, some attention was needed to deal with all the inconsistencies between different pages, requiring some extra steps to extract all needed details of each Resolution.

In the end, all the needed information was combined into a single **Pandas** dataframe, ready to be stored.

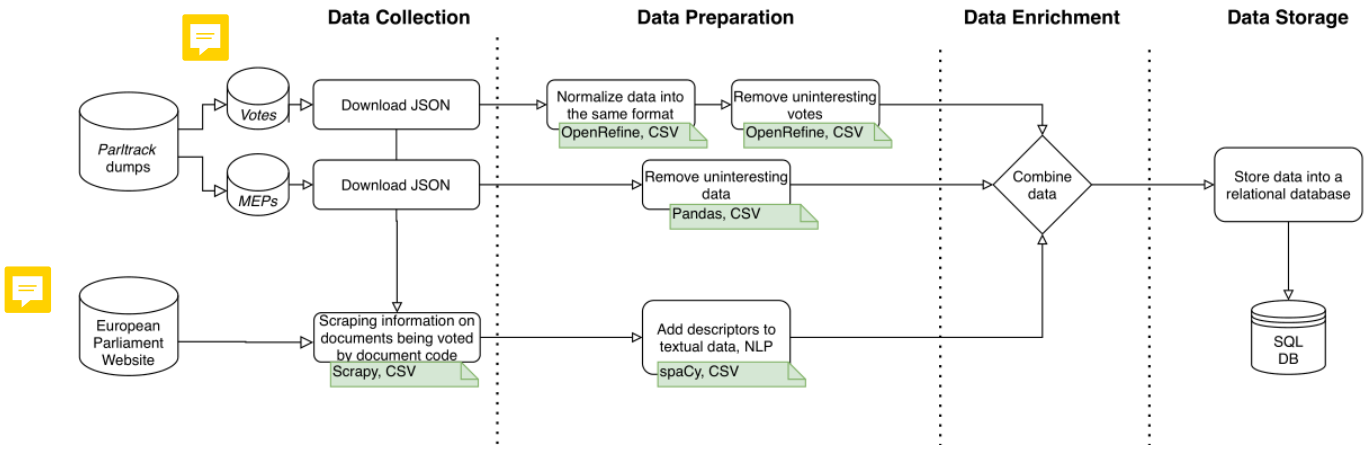

Figure 1: Data Pipeline Diagram

To prepare and add descriptions to textual data of those reports we used **spaCy**. By using spaCy we were able to extract the most important and common keywords present with the reports. Particularly, we applied the language processing pipeline on the reports by joining all the reports' texts, segmenting them into tokens, detecting and labeling named entities.

2.3 Conceptual Model

The main entities of our domain are Committee, Country, **MEPs**, Political Group, **Resolutions** and **Vote** as can be seen in Figure 2.

Each MEP can vote on several **Resolutions** and each can have several **Authors**, or rapporteurs. MEPs can belong to several **Political Groups** since they can change their political affiliation over time, and they are elected to represent a specific **Country**.

Resolutions can be proposed by a **Committee**, while MEPs can belong to several committees as they may change over time.

2.4 Data Storage

In order to make the collected data easily available for the next processing steps, it was decided to store everything in a relational database.

In order for this to be possible, the data needed to be normalized. This means ensuring, for example, that each cell contained only atomic information and that all repeated data was centralized into a single table, thus eliminating all data redundancy.

This required additional processing of some generated **Pandas** dataframes, in order to normalize information. Some data frames were splitted up, with primary/foreign keys being generated to establish a relationship.

The presented conceptual model already accounts for this relational data structure

2.5 System Documents

We will have three documents in our system: **MEPs**, **reports** and **committees**.

The **MEP** document will have a brief bio section of a parliament member's information such as their name, gender, birth date, national party they belong to, their usernames on social media platforms etc. It will also be possible to view the votes cast by them. A user can search for an MEP by their name.

The document **Report** will have a date, a title, content and voting results (total number of votes in favor, against and abstained, and information of who voted). A user can search for resolutions by its title or by keywords present in its content.

The **Committee** document will have the name of the committee and a list of reports motioned by it.

2.6 Collection characterization

2.6.1 ***Reports.*** The information contained in 6396 reports, produced from 2004 to 2020, is being used. We have got this number after heavily filtering all the reports to include only final versions of them. The number of reports per years is very variable but we can see in Figure 3 that there was an unusual low number of reports during the 2009-2014 term.

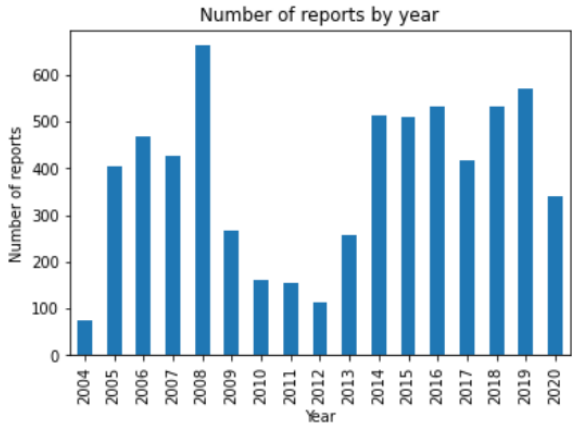

Figure 3: Number of reports per year

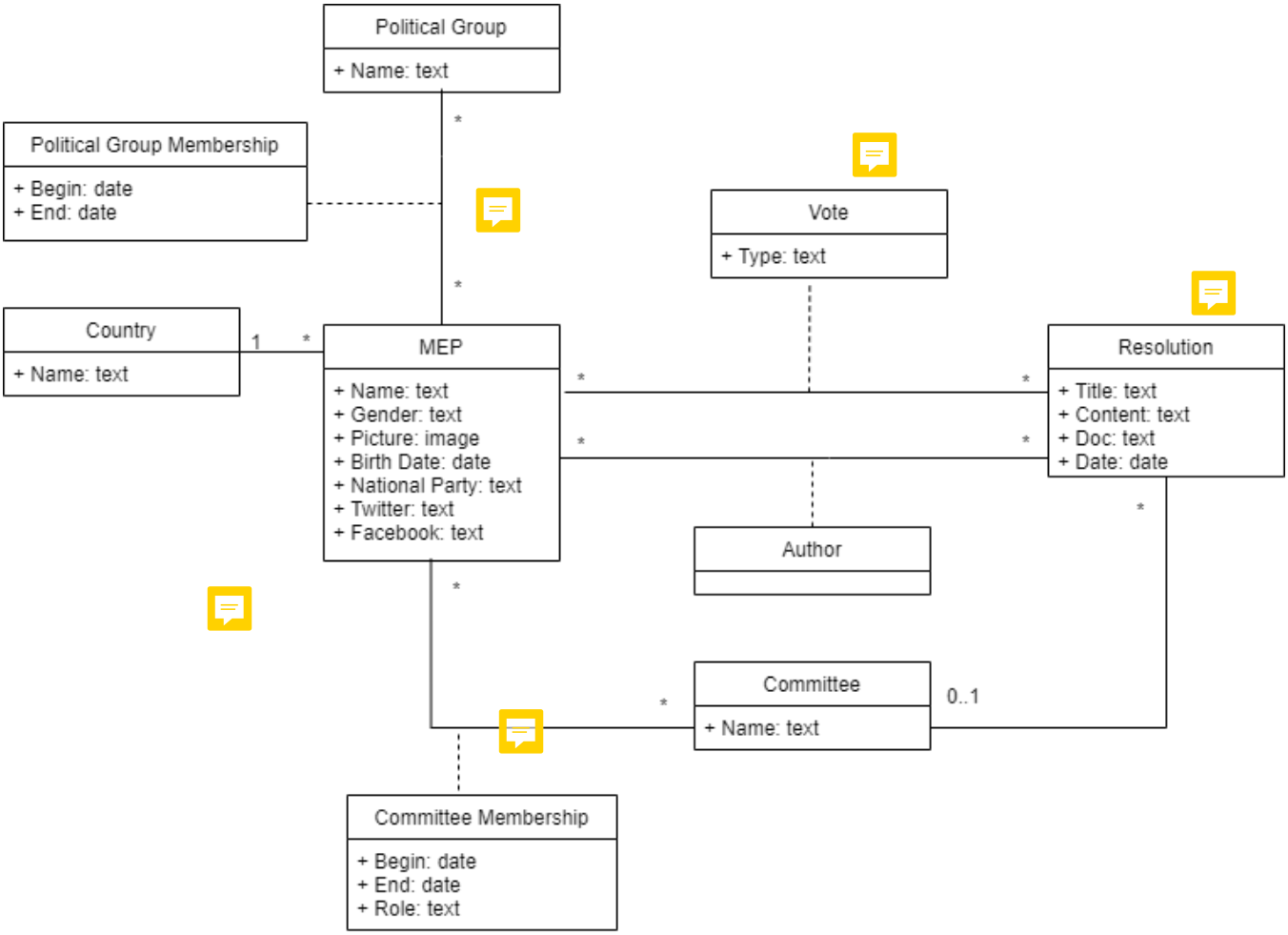

Figure 2: European Parliament's Conceptual Model

The top 9 most common keywords present in the reports are represented in Figure 4. The most used terms are *Council*, *EU* and *Commission*.

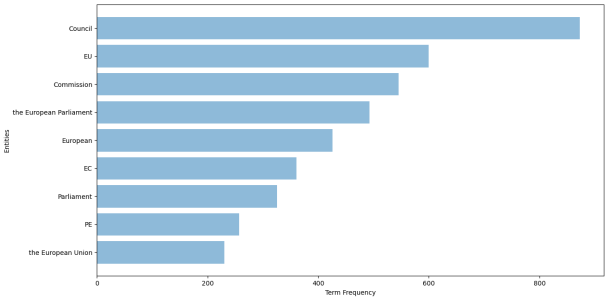

Figure 4: Term frequency of entities in reports

Most of the reports are produced in the context of a committee. There are many however that are produced in other contexts, e.g. written by an individual MEP.

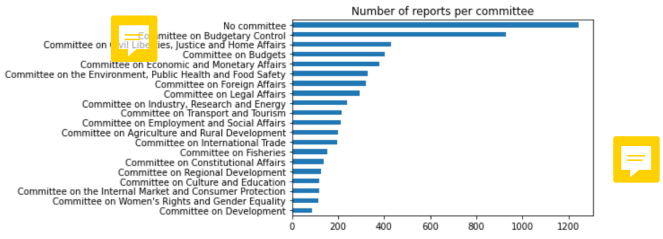

Figure 5: Number of reports per committee

2.6.2 ***Meps***. The information on 4150 MEPs that were at some point part of the parliament from 2004 to 2020 is being used.

The activity of MEPs in terms of votes is very variable. One of the reasons for this is that there are many MEPs that did not spend an entire term - five years - in the parliament. Some of them stayed for a time period inferior to a month, as to replace a temporarily absent MEP for a few weeks.

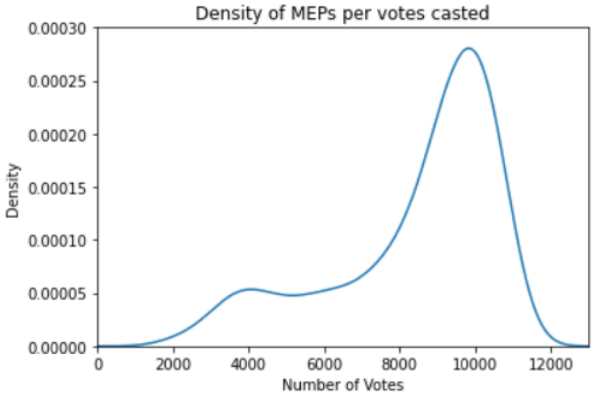

Figure 6: Density of MEPs votes

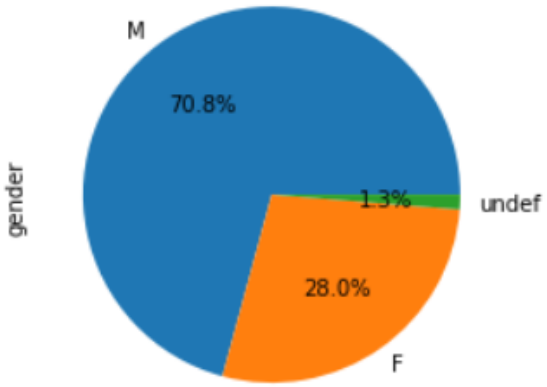

Figure 7: Gender distribution

2.6.3 ***Committees***. The information on 32 committees is being used.

2.7 Data Retrieval tasks

Our results will be focused on the previous documents listed: **MEPs**, **Committees** and **Reports**. Starting with the MEPs, the system will present their personal information, the votes they have cast, political groups they have been part of organized by periods of time and all the committees they have been part of, also organized by period of time. For the committees, the data presented will be about all the reports produced by it and all of its members. Finally, for the reports, the system will present its title and text along with the votes cast for it by MEP and organized by political group.